AN ANIMATED SIMULATOR FOR PACKET SNIFFER

Xiaohong Yuan, Percy Vega, Jinsheng Xu, Huiming Yu, and Stephen Providence

Department of Computer Science, North Carolina A&T State University, 1601 East Market St., Greensboro, NC 27411, Email: {xhyuan, jx, ucshmyu, svp} @ncat.edu; Phone: (336)3347245. [*]*Everett Consulting Corp., 320 McKinley St. #10, Hollywood, FL 33019, Email: percyvega@gmail.com Phone: (954)7325664*

Abstract: Visualization and animation have been used to graphically illustrate various concepts in computer science. This paper describes an animated simulator for packet sniffer. The goal of this tool is to provide users with interactive tutorials and simulations to help them better understand the security concept packet sniffer and related computer networks concepts. This tool can be used in an introductory level computer security course or a computer networks course. It can be used by the instructor in the classroom, and can also be used by the students outside the classroom. This tool will be used in our institute and its effectiveness in teaching and learning will be assessed.

Keywords: Computer science education, security, packet sniffer, animation

1. INTRODUCTION

Visualization has been used in computer science education in the fields of algorithm, computer networks, computer architecture, and operating systems [1-4]. The surveys of computer science educators conducted by the Working Group on "Improving the Educational Impact of Algorithm Visualization" suggest a widespread belief that visualization technology positively impacts learning [5]. It seemed that computing educators are convinced that visualization can make a difference in helping learners better learn the concepts.

It has become increasingly important to provide today's computing students with training and education in security issues. Universities have tried to incorporate security issues into computer science curriculum. In some computer science programs, some security topics are incorporated into existing courses. Others developed new, required courses to the computer science curricula [6, 7, 8]. Resources for teaching computer and information systems security at the undergraduate level will be useful for computer science educators [9].

This paper presents a software tool that demonstrates the security concept of packet sniffer and related computer networks concepts through animation. It is intended to be used in an undergraduate level computer security course or a computer network course. To make this tool more beneficial, exercise problems are designed to improve learner's involvement in learning with this tool [10]. We believe this visualization tool will help students better understand the packet sniffer and network concepts, improve student participation, and make teaching and learning more enjoyable.

2. PACKET SNIFFER

Packet sniffer is a program that captures all of the data packets that pass through a given network interface, and recognizes and decodes certain packets of interest. A packet sniffer is sometimes referred to as a network monitor, or network analyzer. It is normally used by network or system administrator to monitor and troubleshoot network traffic. However, it is sometimes also used by malicious intruders for illicit purpose such as stealing a user's password or credit-card number [11].

Typically, a computer in a network would only capture data packets that are intended for it and ignore packets that are not intended for it. However, if its network interface is configured into promiscuous mode, the computer is capable of capturing all packets traversing the network regardless of the destinations of the packets. A packet sniffer can only capture packets within a given subnet.

There exist various commercial and free packet sniffer tools. Ethereal is an open source tool for troubleshooting, analysis, software and protocol development and education. It can capture data from a live network connection, or read from a capture file. Captured network data can be browsed via a GUI, and more than seven hundred protocols can currently be dissected. A display filter is provided to refine data display [12]. AnalogX PacketMon is a fast and simple to use network monitor that captures IP packets. It uses advanced rules for filter [13]. Network Probe [14] is a tool for traffic-level network monitoring, analysis and visualization. It provides full graphical representation and detailed statistics of the traffic and the type of data traveling across the network. It allows the user to isolate traffic problems and congestions. It also allows the user to search, sort, and filter network traffic information by protocols, hosts, network interfaces, etc. There are yet many other similar packet sniffer tools available.

3. THE PACKET SNIFFER SIMULATOR

The packet sniffer simulator demonstrates visually how a packet sniffer works in a local area network environment, and how data packets are encapsulated and interpreted while going through the protocol stack. The packet sniffer simulator consists of a suite of five demos. Demo I to IV progressively demonstrate how a packet sniffer works. The simulation is based on a network with two subnets. The two subnets are connected with a router, and each subnet has a hub. The first subnet has a star topology and the second subnet has a bus topology. Demo V shows a protocol stack and how a data packet is encapsulated while going down the protocol stack at the source computer, and interpreted while going up the protocol stack at the destination computer.

The learning objectives of this packet sniffer simulator are:

a) Explain the differences between a hub, a bridge/switch, and a router

b) Explain bus and star topology

c) Explain how a data packet is transmitted in a local area network

d) Explain the purpose of "promiscuous mode" of a network interface

e) Explain what a packet sniffer does, and how it works.

f) Explain the encapsulation and de-encapsulation process of a data packet while going through the protocol stack

Macromedia Flash MX Professional Edition was selected to implement the demos of the packet sniffer simulator. The main environment in which Flash animations can run is web page, as a Flash Applet. These animations can also run as a standalone application. The only requirement for both environments is to download and install Macromedia Flash Player which is freely available. The following sections explain the five demos, and the network and security concepts they demonstrate.

3.1 Demo I: Direct Path

Figure 1 shows the user interface of Demo I. Demo II, III, and IV have the same user interface as Figure 1. The user interface can be grouped into four components. The top-left component "Demo Sequence", is a textbox that lists the names of the five demos. It allows the user to select the demo he wants to view. The top-right component "Description Messages", is a text-area that briefly describes the animation. The bottom-right component shows the network architecture. It shows two subnets connected by a router. The subnet to the left has a star topology; the subnet to the right has a bus topology. Each subnet has a hub connecting the computers. The computers are numbered from 0 to 8. The bottom-left component displays the data for simulation. The data used in the simulation is the source address and the destination address of a data packet. Three buttons are provided, which allow user to select from loading default data from an input file, or generating input data randomly, or entering data manually. The input data are displayed in the table that has two columns. Under the first column "From" are source addresses of the data packets. Under the second column "To" are destination addresses of the data packets. Under the two-column data table, are a "play" button (an arrow within a square), and a checkbox "Play continuously". If the checkbox "Play continuously" is checked and the "play" button is clicked, the animation will run from Demo I through Demo IV sequentially and go through all the data pairs in the data table sequentially in each demo. If the "Play continuously" button is not checked and the "play" button is clicked, it will only show one step of the animation, i.e., the animation for one data set. The user can step through the simulation by repeatedly clicking the "play" button. At each step of the simulation, the "description messages" text-area is updated to reflect what is going on.

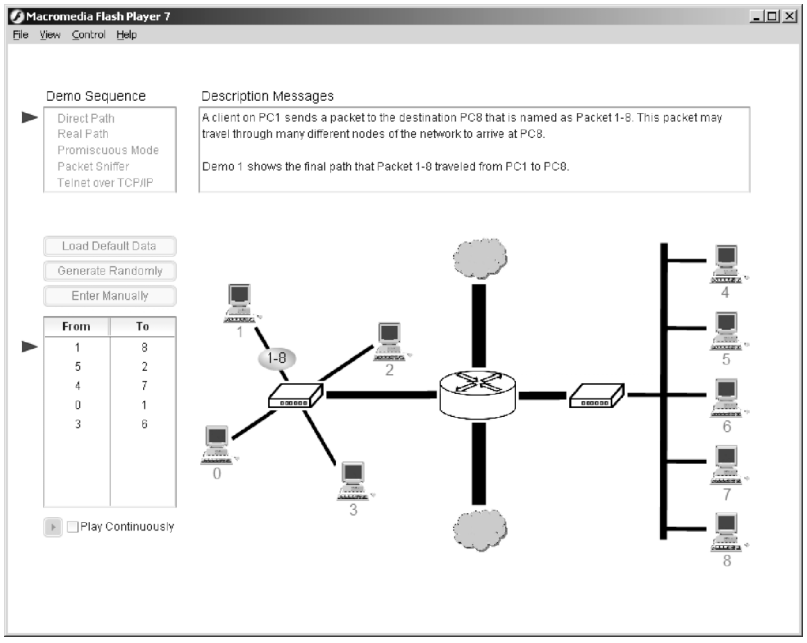

Figure 1. Demo I: Direct Path

Demo I displays the path a data packet from a source goes through to reach the destination. A data packet is represented as an oval shape, labeled by the source and destination numbers. For example, Figure 1 shows a data packet, Packet 1-8 moving from computer 1 to the hub. The packet will go through the hub to the left, the router in the middle, the hub to the right, and finally arrive at computer 8. Keep in mind that, since hub is used in the two subnets, the real path that Packet 1-8 traversed is not the same as the direct path. Demo II demonstrates the real path of a data packet in the simulated network.

3.2 Demo II: The Real Path

Since the computers in the subnet to the left are connected by a hub, all traffic can be seen by all computers on the subnet. The same is true with the subnet to the right which has the bus topology. A network segment that is not switched or bridged (i.e., connected through a hub) is called a common collision domain. Physically, any signal sent across a common collision domain reaches all attached computers. The network interface hardware detects the electrical signal and extracts a copy of the frame. It checks the address of each incoming frame to determine whether it should accept the frame. The network interface hardware compares the destination address in the frame to the computer's physical address (also called hardware address, or media access address). If the destination address in the frame matches the computer's physical address, the interface hardware accepts the frame and passes it to the operating system. If the destination address in the frame does not match the computer's physical address, the interface hardware discards the frame and waits for the next frame to appear [15].

Figure 2 shows that, when computer 1 generates a data packet, the data packet is captured by computer 0, 2 and 3 in the same subnet. Meanwhile, the data packet is forwarded to the router in the middle of the network. The data packet is forwarded to the subnet to the right by the router, and all the computers attached to the bus receive Packet 1-8. Only computer 8's network interface accepts Packet 1-8, and the other computers discard Packet 1-8. This is depicted in Figure 3, in which the comment "mine" pointing to computer 8 indicates that the data packet is accepted by computer 8's network interface, and the comment "not mine" indicates the data packet is discarded by the computer's network interface.

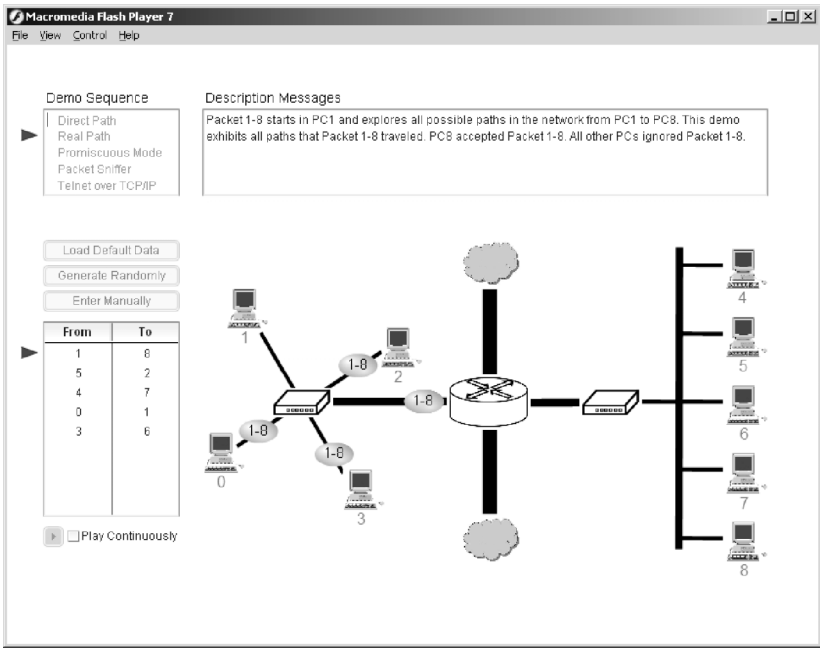

Figure 2. Demo II (a): The Real Path When a Packet is Sent by the Source Computer

If the hub in the subnet to the left is replaced by a switch or bridge, and the computers in the subnet to the right are connected to a switch or bridge too, then the path Packet 1-8 traverses will be the same as Demo I. By comparing Demo I and Demo II, the concepts of repeater, hub, bridge, switch and router can be reviewed.

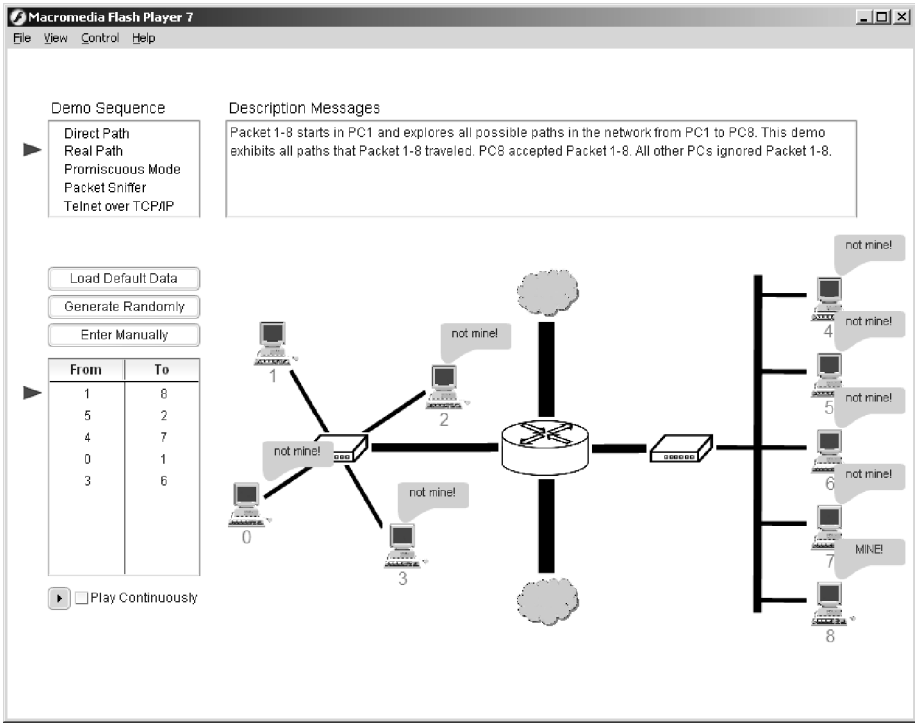

Figure 3. Demo II (b): The Real Path When a Packet is Received by a Destination

A repeater is a hardware device used to extend a network cable. It has two ports. A repeater receives signal from one port, regenerates the signal and sends out to the other port. A hub is a multi-port repeater. A hub receives signal on one port, and regenerates the signal and sends the signal out to all the other ports. Repeaters and hubs work at the Physical layer of the OSI model. A bridge works at the data-link layer of the OSI model. It examines each incoming packet. First the media access address (MAC address) of the sender and the port number through which the packet enters are added into the routing table of the bridge. Then the MAC address of the recipient is looked up from the routing table to determine which port should the packet be forwarded to. If the recipient's MAC address is in the routing table, then the port number is looked up and the packet is forwarded to that port. If the recipient's MAC address is not in the routing table, the bridge sends the signal to all ports except for the one where it was received. A switch is a multi-port bridge. It has the functions of a bridge, but uses a dedicated processor to implement the function, so it is faster than traditional software based bridges. A router works at the network layer of the OSI model. It uses network addresses (for example, IP address) to determine how to forward a packet [16].

3.3 Demo III: Promiscuous Mode

Normally, a computer's network interface hardware checks the destination address of each incoming frame to determine whether it should accept the frame. It discards a frame whose destination address does not match its physical address. However, a computer's network interface hardware can be configured by software into promiscuous mode, which overrides the

conventional address recognition. Once in promiscuous mode, the network interface does not check the destination address, but accepts all frames. The network interface simply places a copy of each frame in the computer's memory and informs the CPU about the arrival of the frame [15].

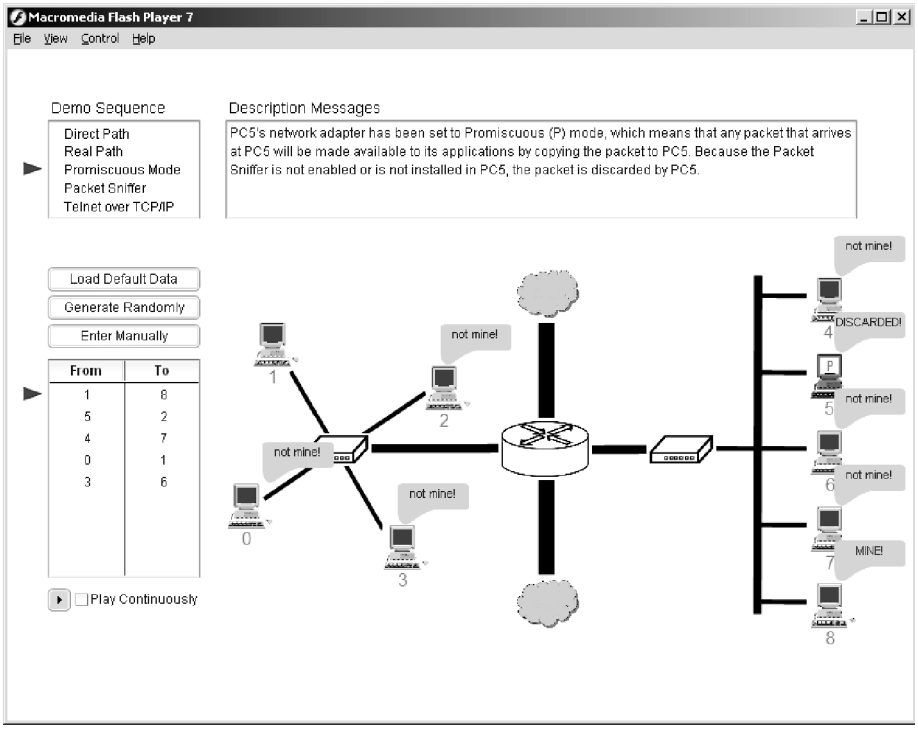

Figure 4. Demo III: Promiscuous Mode

Figure 4 shows the result of an animation of data packet transmission with Computer 5 configured into promiscuous mode. When Packet 1-8 was transmitted on the common bus in the subnet to the right, computer 5 accepted the packet even though the packet was not addressed to it. However, since there was no packet sniffer installed on computer 5 to process the packet, the packet was simply discarded by the operating system of computer 5. Notice the difference between the "DISCARD" comment and the "not mine" comment. In the case of "not mine", the computer's network interface hardware discards the data packet because the destination address is not the same as the physical address of the computer; whereas in the case of "DISCARD", the data packet is accepted by the network interface hardware but is discarded by the operating systems of the computer.

3.4 Demo IV: Packet Sniffer

Figure 5 shows the result of an animation when computer 3 has a packet sniffer installed. The network interface on computer 3 is also configured into promiscuous mode. When Packet 1-8 was sent out to the network, computer 3 captured the package and accepted it, even though the packet was not addressed to computer 3. This is indicated by the "mine" comment in Figure 5. The packet sniffer then examines the content of the data packet according to its configuration. The packet sniffer can be configured by the user to determine what fields to examine and what information it keeps. For example, a packet sniffer can be configured to examine all frames originated from a particular computer, or all TCP/IP traffic, or gather general traffic statistics.

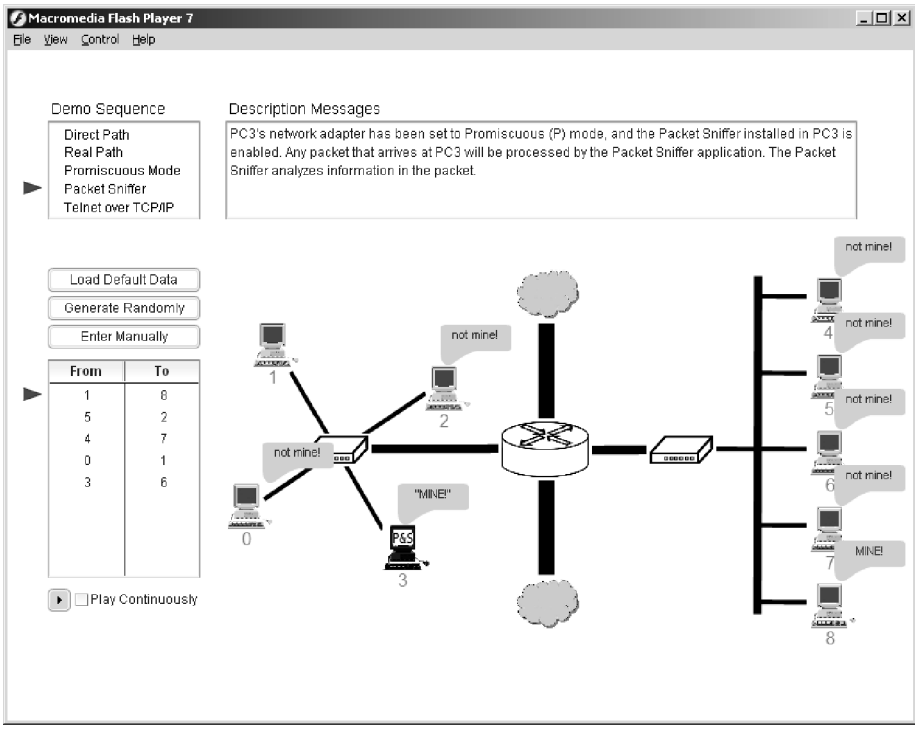

Figure 5. Demo IV: Packet Sniffer

Consider a packet sent from computer 4 to computer 7. When Packet 4-7 reaches the router, it will not be forwarded to the subnet to the left, so computer 3, which has the packet sniffer installed on it, will not be able to capture Packet 4-7. Similarly, if the packet sniffer is installed in one of the computers in the subnet to the right, it would not be able to capture data traffic going through the subnet to the left. A packet sniffer only works in a common collision domain.

3.5 Demo V: Telnet Over TCP/IP

Demo V demonstrates how a data packet is encapsulated and de-encapsulated while going through the protocol stack. It assumes a Telnet application sending data packets over a network with TCP/IP protocol. Figure 6 represents three computers (PC0, PC1 and PC2) connected to a hub. Each computer is represented by a rectangle. In each rectangle a protocol stack of five layers are displayed. The five layers are: application, transport, network, data link and physical layer. The animation shows a data packet generated at the application layer at PC0 being encapsulated while moving down through the protocol stack, and being de-encapsulated while moving up through the protocol stack at PC1 (Figure 7 and Figure 8). The user can step through the animation by clicking the "play" button repeatedly, or run the simulation continuously by checking the checkbox "Play Continuously".

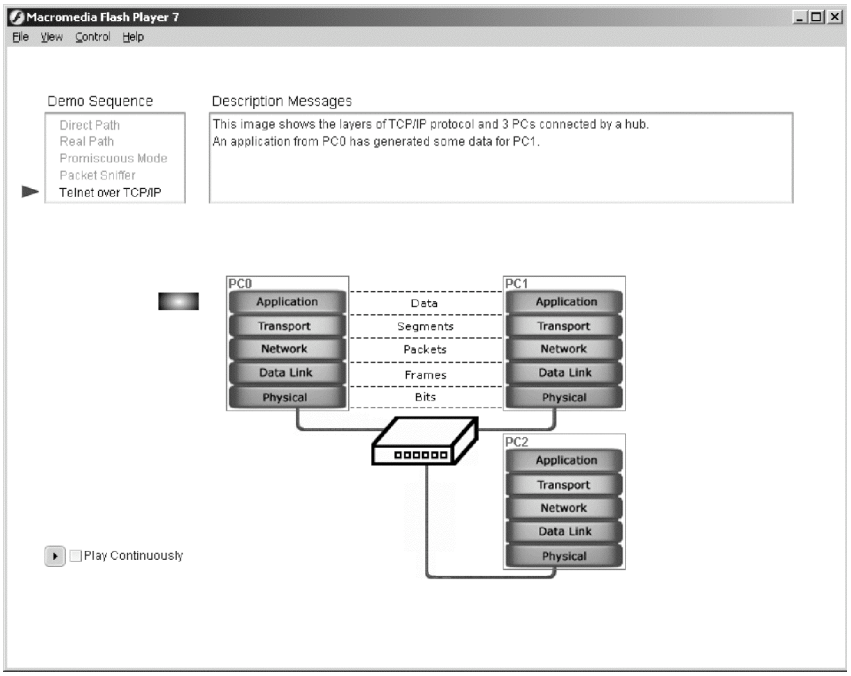

Figure 6. Demo V (a): A Data Packet Generated by PC 0 at Application Layer

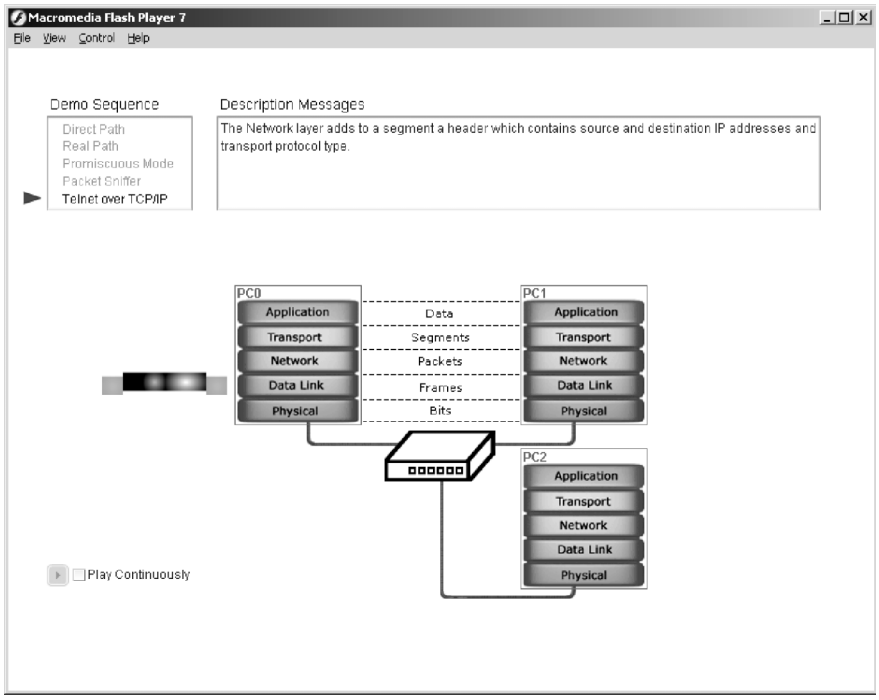

Figure 7. Demo V (b): The Encapsulation Process

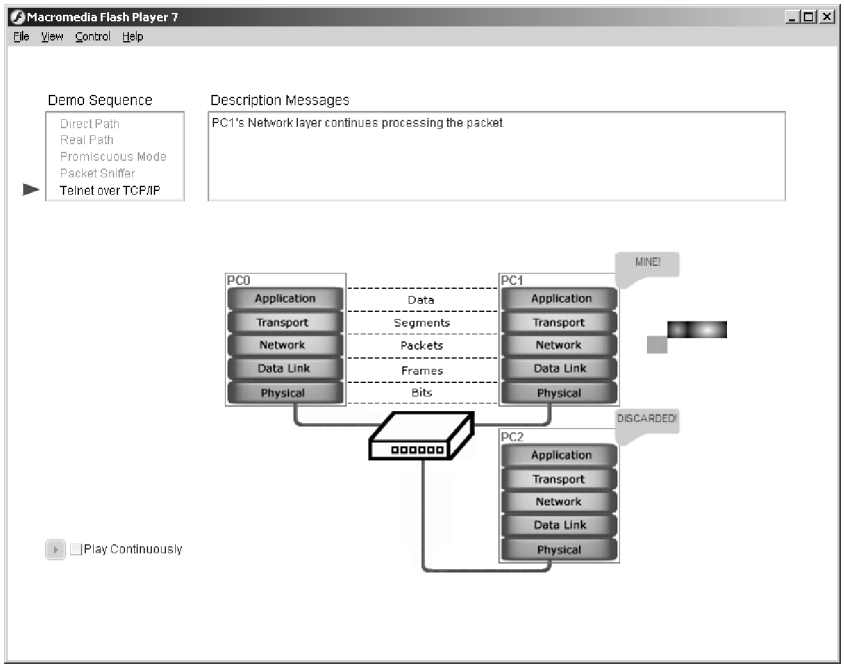

Figure 8. Demo V (c): The De-Encapsulation Process

Encapsulation means that, when the data packet moves down through the protocol layers, a header (and a trailer sometimes) is added at each protocol layer. The transport layer adds a header that contains the source and destination port numbers. The network layer adds a header that contains the source and destination IP addresses and the transport protocol type. The data link layer adds a header and a trailer with source and destination physical addresses and the network type. And the physical layer converts the frame generated by the data link layer into bits. On the destination computer, the de-encapsulation process occurs. The headers (or trailers) are removed in reverse order.

In Figure 8, the data frame is transmitted to PC1 and PC2. At the data link layer, the network interface hardware of PC2 recognizes the destination address of the incoming frame, and finds out it is not addressed to PC2, so PC2 discards the data packet at the data link layer. Whereas PC1's network interface hardware recognizes that the data packet is addressed to PC1. So the frame is forwarded to the operating systems and is further de-encapsulated until it reaches the application layer.

4. RELATED WORK

Holliday [2, 17] developed a series of Java applets and explanatory material to illustrate key computer networking concepts. The applets are: protocol stack applet, error-control applet, reliable data transfer applet and media access applet. The Demo V of our packet sniffer simulator is similar to the protocol stack applet. The protocol stack applet demonstrates how a message goes from the source machine to the destination machine across a router. Whereas Demo V shows how a message goes from the source machine to the destination machine and other machines connected to a hub. Both demonstrate the encapsulation/de-encapsulation process. The protocol stack applet allows the user to specify the sizes of the message and the headers at the different protocol stack layers. The Demo V allows the user to step through the simulation or run the simulation continuously.

White [18] developed a set of visualization tools for teaching a data communications and computer networks course. A collection of eleven computer based training modules was created as an educational supplement for a textbook on data communications and computer networks. Some of the modules demonstrate similar networking concepts as our packet sniffer simulator. For example, Module 1 demonstrates the concept of encapsulation and de-encapsulation; Module 6, 7 and 8 demonstrates the basic concepts of local area networks, and simulate internetworking with hubs, bridges and switches.

The "increasing security in aviation-oriented computing education: a modular approach" project at Embry-Riddle Aeronautical University [19, 20] is developing interactive modules for several topics: buffer overflow vulnerabilities, cryptography, and multimodal transportation and Bioterrorism defense. These interactive modules use software simulation and visualization to demonstrate security concepts. They may be used by an instructor for classroom or laboratory work. Our packet sniffer simulator can be considered as an interactive module with similar intention, demonstrating the packet sniffing vulnerability in a network environment.

CyberCIEGE [21] is a high-end, commercial-quality video game developed for teaching security concepts and practices. It is a resource-management simulation in which the players engage in planning and construction and observe the results of their choices.

5. CONCLUSION AND FUTURE WORK

The packet sniffer animator demonstrates the packet sniffer and local area network concepts through five demos. It can be used by instructors of computer networks and security in the classroom. Tutorials and exercises are designed with the tool to help students to gain deep understanding of packet sniffer and related network concepts. In [5], six forms of learner engagement with visualization technology are defined, they are: 1) no viewing; 2) viewing; 3) Responding; 4) Changing; 5) Constructing; and 6) Presenting. We can design exercises so that the students can be engaged in the forms of responding and changing. The packet sniffer animator can be accessed at:

http://clayton.ncat.edu/comp476/PacketSnifferAnimation/index.html

We plan to use this tool in an undergraduate level course Computer Networks, and an Applied Network Security course in the Fall 2005 semester. The effectiveness of the tool in teaching and learning will be assessed in these two courses. Questionnaire will be designed to collect students' opinion on this tool. The future work will also include the development of animation tools for other more complicated security concepts, for example, the Kerberos architecture and distributed denial of services, and the assessment of these tools.

The CAPE and eLMS [22] developed at Vanderbilt University's Institute for Software Integrated Systems are a technology infrastructure for adaptive on-line learning. CAPE is used to design how learning materials are used to create an adaptive learning experience, and eLMS is a web-based delivery platform. We are interested in investigating the possibility of using CAPE and eLMS to extend the Packet Sniffer simulator to create adaptive learning experiences for computer network and computer security courses.

ACKNOWLEDGEMENT

This work was supported by National Science Foundation under the award number DUE-0415571.

REFERENCES

1. Harrold, M. J. and Stasko, J. Algorithm animation, 2002.Available at: http://www.cc.gatech.edu/gvu/softviz/algoanim/
2. Holliday, M. A. Animation of computer networking concepts, ACM Journal of Educational Resources in Computing, Vol. 3, No. 2, June 2003, Article 2.
3. Null, L. and Rao, K. CAMERA: Introducing memory concepts via visualization, Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, USA, February 23-27, 2005, pg. 96-100.
4. Carr, S., Mayo, J. and Shene, C.K. ThreadMentor – A system for teaching multithreaded programming, 2003. Available at: http://www.cs.mtu.edu/~shene/NSF-3/
5. Naps, T. L. et. al. Exploring the role of visualization and engagement in computer science education, ACM SIGCSE Bulletin, Vol. 35, Issue 2, 2003, pg. 131-152.
6. Frincke, D. and Bishop M. Joining the security education community, IEEE Security and Privacy, Vol. 2, Issue 5, 2004, pg. 61-63.
7. LeBlanc, C. and Stiller E. Teaching computer security at a small college. Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk, Virginia, USA, March 3-7, 2004, pg. 407-411.
8. Mullins, P. et. al. Panel on integrating security concepts into existing computer courses, Proceedings of the 33th SIGCSE Technical Symposium on Computer Science Education, Nrothern Kentucky, Cincinnati, USA, February 27-March 3, 2002, pg. 365-366.
9. Bhagyavati, et. al. Teaching hands-on computer and information systems security despite limited resources, Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education, St. Louis, Missouri, USA, February 23-27, 2005, pg. 325-326.
10. Grissom, S. et. al. Algorithm visualization in CS Education: Comparing levels of student engagement, Proceedings of ACM 2003 Symposium on Software Visualization, San Diego, California, , USA, June 11-13, pg. 87- 94.
11. Bradley, etc. "Introduction to Packet Sniffing", 2005. Available at: http://netsecurity.about.com/cs/hackertools/a/aa121403.htm
12. Combs, G. Ethereal, 2005. Available at: http://www.ethereal.com/
13. AnalogX, PacketMon, 1998-2003. Available at: http://www.analogx.com/contents/download/network/pmon.htm
14. Objectplanet, Network Probe, 2005. Available at: http://www.objectplanet.com/probe/
15. Comer, D. E. Computer Networks and Internets, 4th edition, Pearson Prentice Hall, 2004.
16. Shields, P. Networks in-depth: switches, hubs and routers, 2005. Available at: http://www.thebusinessmac.com/features/network_indepth.shtml
17. Holliday, M. A. and Johnson, M. "A Web-Based Introduction to Computer Networks for Non-Majors - The Protocol Stack", February, 2004. Available at: http://cs.wcu.edu/~holliday/cware/Stack/indexStack.html
18. White, C. M. Visualization tools to support data communications and computer network courses", Journal of Computing Sciences in Colleges, Vol. 17, Issue 1, pg. 81-89.
19. Crandall, J.R., et. al. Driving home the buffer overflow problem: a training module for programmers and managers, Proceedings of National Colloquium for Information Systems Security Education (NCISSE 2002), Washington, 2002.
20. Gerhart, S. et. al. Increasing security in aviation-oriented computing education: a modular approach, August 2005. Available at: http://nsfsecurity.pr.erau.edu/
21. Irvine, C. E. and Thompson, M. F. "CyberCIEGE: Gaming for Information Assurance", IEEE Security and Privacy, Volume 3, Issue 3, pg. 61-64.
22. Sztipanovits, J. et al. "Introducing Embedded Software and Systems Education and Advanced Learning Technology in Engineering Curriculum", ACM Transactions of Embedded Computing Systems, Special Issue on Education, 2005.